



# Алгоритмы и структуры данных

## Лекция 4. Структуры данных и их классификация

---

Антон Штанюк (к.т.н, доцент)

5 февраля 2021 г.

Нижегородский государственный технический университет им. Р.Е. Алексеева  
Институт радиоэлектроники информационных технологий  
Кафедра "Компьютерные технологии в проектировании и производстве"

Классификация структур данных

Операции над СД

Список литературы

## Цитата 1

*Плохие программисты думают о коде. Хорошие программисты думают о структурах данных и их взаимосвязях», — Линус Торвальдс, создатель Linux.*

## Цитата 2

*«Алгоритмы + структуры данных = программы». - Н. Вирт*

**Структура данных (СД)** - это программная абстракция, позволяющая хранить и обрабатывать данные, представленные в определенном виде. Структура данных неразрывно связана с набором операций, которые можно к этим данным применять. Структура должна обеспечивать удобство и скорость применения определенных операций. Связь между структурой хранения данных и операциями формально закрепляется в языках программирования с помощью понятия *тип данных*.

Математически, структуру данных можно определить как совокупность трех множеств  $\langle I, O, R \rangle$ , где

- $I$  - множество элементов.
- $O$  - множество операций
- $R$  - множество отношений между элементами.

## Классификация структур данных

---

Все структуры данных можно разделить на две группы:

1. Примитивные (скалярные)
2. Составные (векторные)

К примитивным структурам относятся традиционные (классические) типы данных, реализованные в различных языках программирования и состоящие из *битов*. Сюда можно отнести *числа, символы, логические значения, значения адреса, перечисления*. При реализации в разных языках программирования могут быть свои особенности. В языке Си, например, символы и перечисления реализуются через числовые типы.

В рамках нашего курса наибольший интерес вызывают **составные** структуры, в которых составными частями являются другие структуры данных.



Следует различать **физическую** и **логическую** структуру данных. Когда говорят о физической структуре простых типов, то под этим понимают их размер и способ размещения битов в памяти. С точки зрения логической структуры простой тип является неделимой элементарной единицей.

В дальнейшем, чтобы подчеркнуть именно физический характер СД, мы будем использовать термин **структура хранения**.

**Изменчивостью** структуры называется изменение числа элементов и связей между ними.

В соответствии с признаком изменчивости структуры делятся на

1. **Статические** (массив, запись, множество, битовое поле)
2. **Динамические** (связанный список, дерево)

Основным отличием статических структур от динамических, является распределение памяти и, соответственно, изменение числа элементов. Для статических структур память отводится заблаговременно и не меняется в процессе работы программы. Для динамических структур память отводится «на ходу» в процессе выполнения программы. Для работы с динамическими типами данных в разных языках программирования широко используются указатели.

По расположению в памяти выделяют структуры **внутренней** и **внешней** памяти. К СД внешней памяти относят **файлы**.

Элементы в структуре могут быть **упорядоченными** и **неупорядоченными**. В соответствии с этим признаком структуры делятся на

1. Нелинейные (деревья, графы, сети)
2. Линейные (вектор, строка, массив, стек, очередь, список);

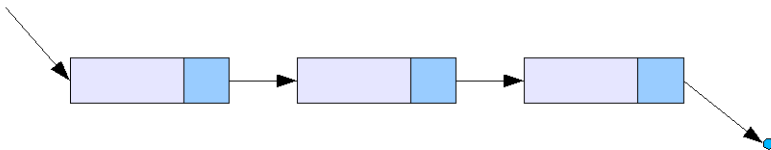
В линейной коллекции можно выделить первый, второй, ..., последний элементы. В нелинейной коллекции такой упорядоченности нет.

К фундаментальным структурам можно отнести:

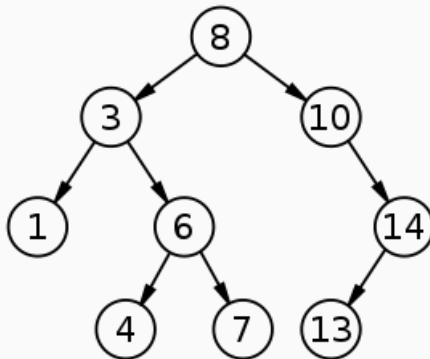
- **Массив** - набор из однотипных данных, занимающих смежную память.



- **Связанный список** - набор из однотипных данных, линейно связанных друг с другом посредством ссылок (указателей).



- **Дерево** - набор из однотипных данных, нелинейно связанных друг с другом посредством ссылок (указателей).





- **Запись** - набор из разнотипных данных, занимающих смежную память.



К структурам, которые могут быть созданы на основе фундаментальных можно отнести:

1. **Многомерный массив** - на основе обычного массива.
2. **Вектор** - на основе массива.
3. **Хеш-таблица (ассоциативный массив)** - на основе массива и списка (массив списков).

Некоторые структуры данных не имеют конкретной реализации и могут создаваться за счет разных структур. К ним относятся:

1. **Стек** - на основе массива или списка.
2. **Очередь** - на основе массива или списка.
3. **Дек** - на основе массивов или списка.
4. **Множество** - на основе хеш-таблицы или дерева.
5. **Словарь** (отображение) - на основе хеш-таблицы или дерева.

## Операции над СД

---

К числу основных операций над структурами данных можно отнести:

1. Вставка элемента.
2. Удаление элемента.
3. Доступ к элементу.
4. Поиск элемента.

Рассмотрим асимптотическую сложность операций, применительно к различным структурам данных.

| Операция | Сложность (средняя) | Сложность (наихудшая) |
|----------|---------------------|-----------------------|
| Доступ   | $O(1)$              | $O(1)$                |
| Поиск    | $O(n)$              | $O(n)$                |
| Вставка  | $O(n)$              | $O(n)$                |
| Удаление | $O(n)$              | $O(n)$                |

| Операция | Сложность (средняя) | Сложность (наихудшая) |
|----------|---------------------|-----------------------|
| Доступ   | $O(n)$              | $O(n)$                |
| Поиск    | $O(n)$              | $O(n)$                |
| Вставка  | $O(1)$              | $O(1)$                |
| Удаление | $O(1)$              | $O(1)$                |

| Операция | Сложность (средняя) | Сложность (наихудшая) |
|----------|---------------------|-----------------------|
| Доступ   | $O(\log_2 n)$       | $O(n)$                |
| Поиск    | $O(\log_2 n)$       | $O(n)$                |
| Вставка  | $O(\log_2 n)$       | $O(n)$                |
| Удаление | $O(\log_2 n)$       | $O(n)$                |








## Дерево (сбалансированное)






| Операция | Сложность (средняя) | Сложность (наихудшая) |
|----------|---------------------|-----------------------|
| Доступ   | $O(\log_2 n)$       | $O(\log_2 n)$         |
| Поиск    | $O(\log_2 n)$       | $O(\log_2 n)$         |
| Вставка  | $O(\log_2 n)$       | $O(\log_2 n)$         |
| Удаление | $O(\log_2 n)$       | $O(\log_2 n)$         |

| Операция | Сложность (средняя) | Сложность (наихудшая) |
|----------|---------------------|-----------------------|
| Доступ   | $O(1)$              | $O(n)$                |
| Поиск    | $O(1)$              | $O(n)$                |
| Вставка  | $O(1)$              | $O(n)$                |
| Удаление | $O(1)$              | $O(n)$                |

## Список литературы

---

-  Кормен Т., Лейзерсон Ч., Ривест Р.  
*Алгоритмы: построение и анализ*  
МЦНМО, Москва, 2000
-  Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы:  
построение и анализ.  
*2-е изд. — М.: «Вильямс», 2006*
-  Википедия  
*Алгоритм*  
<http://ru.wikipedia.org/wiki/Алгоритм>
-  Википедия  
*Список алгоритмов*  
[http://ru.wikipedia.org/wiki/Список\\_алгоритмов](http://ru.wikipedia.org/wiki/Список_алгоритмов)
-  Традиция  
*Задача коммивояжёра*  
<http://traditio.ru/wiki/Задача>

-  Википедия  
*NP-полная задача*  
<http://ru.wikipedia.org/wiki/NP-полная>
-  Серджвик Р.  
*Фундаментальные алгоритмы на C++. Части 1-4*  
Diasoft, 2001
-  Седжвик Р.  
*Фундаментальные алгоритмы на C. Анализ/Структуры данных/Сортировка/Поиск*  
СПб.: ДиаСофтЮП, 2003
-  Седжвик Р.  
*Фундаментальные алгоритмы на C. Алгоритмы на графах*  
СПб.: ДиаСофтЮП, 2003
-  Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы.  
*Издательский дом «Вильямс», 2000*



Кнут Д.

*Искусство программирования, том 1. Основные алгоритмы*  
3-е изд. — М.: «Вильямс», 2006



Кнут Д.

*Искусство программирования, том 2. Получисленные методы*  
3-е изд. — М.: «Вильямс», 2007



Кнут Д.

*Искусство программирования, том 3. Сортировка и поиск*  
2-е изд. — М.: «Вильямс», 2007



Кнут Д.

*Искусство программирования, том 4, выпуск 3. Генерация всех сочетаний и разбиений*  
М.: «Вильямс», 2007



Кнут Д.

*Искусство программирования, том 4, выпуск 4. Генерация всех деревьев. История комбинаторной генерации*

М.: «Вильямс», 2007